Everyday Practices of Agile Software Developers

Julia Prior

A Thesis submitted for the Degree of Doctor of Philosophy in Computing Sciences

Faculty of Engineering and Information Technology
University of Technology, Sydney

July 2011

Certificate of Authorship/Originality

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student

Acknowledgments

Producing a doctoral thesis is not an endeavour that I could have accomplished alone. Many people contributed to the effort in myriad ways. To the numerous friends, colleagues and family members who cheered me on from the sidelines and provided other valuable support, I thank you. While it is not feasible to acknowledge everyone, there are some contributions that must be specifically mentioned.

Firstly, an enormous thank you to the participant developers at Raptor who so generously allowed me into their working lives, as a stranger and novice, enthusiastically and patiently sharing their daily experiences with me. The CEO was exceptional in his enthusiasm, support and generosity for this research, giving me access to whomever and whatever I deemed necessary to carry out my fieldwork. I am extremely grateful to them all. I trust that the stories of their work in this ethnography and the findings of the research will make them proud to have played such a crucial part.

I was privileged to share the joys and despairs of the PhD journey with several exceptional students: Thomas Given-Wilson offered lively intellectual debates and challenging critiques of several ideas and chapter drafts; fellow doctoral candidates in the IDHuP research laboratory, Lian Loke, Penny Hagen, Kirsten Sadler, Astrid Larssen and Dean Hargreaves, in particular, helped me work through new ways of thinking about knowledge and our understanding of the world, as well as giving pragmatic moral support at pivotal moments; and my dear friend, Gordana Culjak — as we travelled the part-time PhD journey together over several years, albeit in very different types of research, her understanding, encouragement and advice helped me over many a hump in the road. I thank you all.

I also thank the other members and associates of the IDHuP lab for contributing to and being part of an holistic research space. I thank Tim Mansfield for his role as my acting supervisor in the middle of my candi-

dature particularly, but also for his advice, inspiration and example over many years. Thank you, too, to Ken Dovey, for his valued advice and encouragement throughout this project.

Richard Raban, former Head of the School of Software, viewed the completion of my thesis as vital for my own professional and personal development, as well as to the school and the faculty. My appreciation to him for continually insisting that it was a priority for me and providing managerial support and space to facilitate this project is immeasurable. Thank you.

I regard it as a tremendous honour to have been supervised in this research by two remarkable people. Toni Robertson, my principal supervisor, turned my view of the world upside down and inside out. She inspired and challenged me with interesting, but often daunting, suggestions and ideas to grapple with and explore, in order to engage more profoundly with the world's elusive, dynamic and intriguing complexities. And she never let me forget the fundamental importance to research rigour and integrity of the researcher's own critical, explicit positioning and accountability.

John Leaney, my mentor and co-supervisor, enthusiastically and generously embraced my proposed research direction from the first. His keen interest and belief in the value of the research never waned and set me back on track with renewed purpose, time and again.

Toni and John opened up alternative approaches to research in general, and software engineering practice in particular, that I had either not previously been exposed to or had not realised were scholarly options. They were dedicated to and engaged with me in this research process at a level that seems to be only dreamed of by most PhD students for their supervision. Thank you both for your commitment, your example and your unflagging belief in me.

My beloved parents, Mike Coleman and Niki Sampson, instilled in me from an early age an insatiable curiosity about the world, a love of study and a passion for the written word, thus preparing me for this particular learning journey. I am very grateful to my mother for casting her careful and professional eye over two drafts of my thesis, and for all her helpful suggestions.

My precious sons, Matthew and Luke, despite waiting patiently for years of their childhoods for bigger slices of my time and attention, showed great pride in my efforts throughout. Thank you both for providing perspective and laughter at all the right moments.

My husband and hero, Andrew, supported me through this endeavour for much longer than anyone should be expected to do. Throughout, he believed that it was only a matter of time before I was done, and he cheerfully shouldered the bulk of the domestic load to enable me to study. Whatever I say here to express my appreciation to him will be completely inadequate, so I will simply say thank you, from the bottom of my heart.

Contents

List of Figures

Abstract

The theory and practice of software engineering is largely represented in the literature by discrete rational models, methodologies, tools and techniques for producing software and for describing work processes as they ought to be done. There has been little research that characterises software development as it is actually practised every day by professional software developers. This thesis addresses that gap.

As the recognition of software development as embodied work is fundamental to this research, a situated approach underpins its exploration of software development as it is performed by real developers in their workplace at the time that it occurs. The initial, guiding research question,
"So what **do** *software developers do every day?"*
was addressed by undertaking a longitudinal ethnographic study, involving two years of fieldwork, of an Australian group of software developers who use an Agile development methodology.

A local software development infrastructure, comprising software tools and formal software development policies and procedures, is available to the developers. As the research progressed, it became obvious that they spent an enormous amount of their time and effort dealing with this infrastructure in order to accomplish their daily work. Further, more directed questions about the developers' daily practice thus became distilled into this one:
What significance does their infrastructure have for the developers' everyday work?

The findings challenge two assumptions often implicit in software engineering research interests: firstly, that infrastructure, along with its constituent elements, is a given — a discrete, independent and stable entity — and, secondly, that software development practice can be authentically represented by, and understood in, formal, rational models of software process. The foregrounding of relations between the developers, their infrastructure and their work product forms the crux of this thesis. Its primary contributions are:

- It presents an ethnography of a little-studied group: professional software developers;

- It provides a detailed account of how the developers' infrastructure and their work practice are intimately intertwined and mutually constitutive, continuously generating the context in which the design and development of complex software can be accomplished;
- It articulates the processes and practices that effect progressive blurring and co-dependency between the software product and the infrastructure;
- It identifies and describes essential dimensions of developers' practice that are missing from accounts and representations such as formal models;
- It demonstrates how an explication of local infrastructure is more beneficial to understanding specific software development work practice than abstract definitions of infrastructure which are not constrained by the practicalities of everyday use.